Comput Optim Appl (2015) 60:413–422
DOI 10.1007/s10589-014-9678-4

Limited memory BFGS method based on a high-order tensor model

Fahimeh Biglari · Ali Ebadian

Received: 16 July 2010 / Published online: 24 July 2014

Abstract This paper is aimed to employ a modified quasi-Newton equation in the framework of the limited memory BFGS method to solve large-scale unconstrained optimization problems. The modified secant equation is derived by means of a forth order tensor model to improve the curvature information of the objective function. The global and local convergence properties of the modified LBFGS method, on uniformly convex problems are also studied. The numerical results indicate that the proposed limited memory method is superior to the standard LBFGS method.

Keywords Large scale nonlinear optimization · Modified limited memory quasi-Newton method · Curvature approximation · Modified quasi-Newton equation · R-linear convergence

Mathematics Subject Classification 90C06 · 90C53

1 Introduction

In this paper we describe a limited memory quasi Newton (QN) algorithm for solving large nonlinear optimization problems. We write this problem as

$$\min f(x),$$

F. Biglari (✉)
Department of Mathematics, Faculty of Science, Urmia University of Technology,
Urmia, West Azarbayjan, Iran
e-mail: fbeiglari@gmail.com

A. Ebadian
Department of Mathematics, Faculty of Science, Urmia University, Urmia, West Azarbayjan, Iran
e-mail: ebadian.ali@gmail.com

where $f : \mathbb{R}^n \to \mathbb{R}$ is a nonlinear and twice continuously differentiable function and the number of variables n is assumed to be large.

Problems with the size of thousands can be solved efficiently, only if the storage and computational cost of the optimization algorithm can be kept at a tolerable level. Among the methods for solving these problems are inexact Newton methods, sparse elimination techniques, limited memory QN methods and conjugate gradient methods [5]. In this paper the focus is directed toward the limited memory quasi-Newton methods.

The limited memory QN methods continuously refresh the correction set of small m positive integer vector pairs by removing and adding information at each iteration. The resulting vector pairs contain curvature information from only the m most recent iterations to construct Hessian approximation, so that the main properties of QN updates are maintained. These methods are robust, inexpensive and easy to implement and due to the low storage demand, they are suitable for large problems, see [5].

To improve the efficiency properties of the limited memory algorithm, one needs to achieve more accurate Hessian approximation at lower cost in computation and storage. As regards, the main contribution of this paper is to use a fourth order tensor model to derive a QN equation which enables us to modify the limited memory BFGS method.

In the following section, we describe derivation of the modified secant equation. The global and local convergence result of the modified limited memory BFGS method is presented in Sect. 3. In Sect. 4, we will describe the detailed algorithm of modified limited memory BFGS, Algorithm 2, based upon two loop recursive procedure. In Sect. 5, we compare the Dolan and Moré [2] performance profile of the new algorithm with LBFGS by Liu and Nocedal [4]. The conclusion is finally outlined in Sect. 6.

2 Derivation of modified secant equation

In this section we derive a more efficient QN equation by taking the forth order tensor model into account. The general idea can be delineated as follows: Recall that the ordinary QN method generates a sequence $\{x_k\}$, at iterate k, by

$$x_k = x_{k-1} - \alpha_{k-1} H_{k-1} g_{k-1},$$

where α_{k-1} is the stepsize, g_{k-1} denotes the gradient of f at x_{k-1}, and H_{k-1} is updated at every iteration to obtain H_k that satisfies the secant equation

$$H_k y_{k-1} = s_{k-1}, \tag{1}$$

in which $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. Based on the fourth order tensor model, provided that $s_{k-1}^T y_{k-1} \neq 0$, the secant equation can be modified in the following form

$$H_k \widetilde{y}_{k-1} = s_{k-1}, \qquad \widetilde{y}_{k-1} = \left(1 + \frac{\phi_{k-1}}{s_{k-1}^T y_{k-1}}\right) y_{k-1}, \tag{2}$$

where $\phi_{k-1} = 4(f_{k-1} - f_k) + 2(g_k + g_{k-1})^T s_{k-1}$. Furthermore, it is proved that if f is smooth enough, and $\|s_{k-1}\|$ is sufficiently small, then

$$s_{k-1}^T G_k s_{k-1} - s_{k-1}^T \widetilde{y}_{k-1} = \frac{1}{6} s_{k-1}^T (T_k s_{k-1}) s_{k-1} + O(\|s_{k-1}\|^5), \tag{3}$$

where T_k is three-dimensional tensor of f at x_k.

We produce the steps, that lead to a new modified secant equation as below: consider the fourth-order tensor model of the objective function about the iterate x_k:

$$f(x_k + s) = f_k + g_k.s + \frac{1}{2} G_k.s^2 + \frac{1}{6} T_k.s^3 + \frac{1}{24} V_k.s^4 + O(\|s\|^5), \tag{4}$$

where V_k is a four-dimensional tensor of f at x_k, see [6]. Substituting $s = -s_{k-1}$ into (4), it follows that

$$f_{k-1} = f_k - g_k.s_{k-1} + \frac{1}{2} G_k.s_{k-1}^2 - \frac{1}{6} T_k.s_{k-1}^3 + \frac{1}{24} V_k.s_{k-1}^4 + O(\|s_{k-1}\|^5). \tag{5}$$

Differentiate (4) at $s = s_{k-1}$, and multiply both sides of the outcome by $-s_{k-1}$, yields

$$g_{k-1}.s_{k-1} = g_k.s_{k-1} - G_k.s_{k-1}^2 + \frac{1}{2} T_k.s_{k-1}^3 - \frac{1}{6} V_k.s_{k-1}^4 + O(\|s_{k-1}\|^5). \tag{6}$$

After multiplying (5) by 4, and adding (6), the terms involving the tensor V_k is eliminated, and we get

$$s_{k-1}^T G_k s_{k-1} = y_{k-1}^T s_{k-1} + \phi_{k-1} + \frac{1}{6} s_{k-1}^T (T_k s_{k-1}) s_{k-1} + O(\|s_{k-1}\|^5). \tag{7}$$

This ensures that with respect to the expression (2), the last relation is equivalent to (3). On the other direction, it can be readily seen that

$$s_{k-1}^T G_k s_{k-1} = s_{k-1}^T y_{k-1} + \phi_{k-1} + O(\|s_{k-1}\|^3).$$

This result suggests that the curvature $s_{k-1}^T \widetilde{y}_{k-1}$ captures the second order curvature $s_{k-1}^T G_k s_{k-1}$ with a higher precision (lower error). This high precision can be achieved with essentially no additional computational effort.

The modified QN Eq. (2) leads us to modify the inverse BFGS update formula, namely MBFGS update, as follows

$$\begin{aligned} H_k &= \left(I - \frac{s_{k-1} \widetilde{y}_{k-1}^T}{\widetilde{y}_{k-1}^T s_{k-1}} \right) H_{k-1} \left(I - \frac{\widetilde{y}_{k-1} s_{k-1}^T}{\widetilde{y}_{k-1}^T s_{k-1}} \right) + \frac{s_{k-1} s_{k-1}^T}{\widetilde{y}_{k-1}^T s_{k-1}} \\ &= V_{k-1}^T H_{k-1} V_{k-1} + \frac{\rho_{k-1}}{\beta_{k-1}} s_{k-1} s_{k-1}^T, \end{aligned} \tag{8}$$

where $\rho_{k-1} = 1/y_{k-1}^T s_{k-1}$, $\beta_{k-1} = 1 + \phi_{k-1} \rho_{k-1}$ and $V_{k-1} = I - \rho_{k-1} y_{k-1} s_{k-1}^T$.

Formula (8) maintains the positive definiteness of H_k if

$$s_{k-1}^T y_{k-1} > 0 \quad \text{and} \quad s_{k-1}^T \widetilde{y}_{k-1} > 0.$$

The left hand side inequality is guaranteed, since the steplength α_{k-1} is computed such that the following Wolfe conditions hold:

$$f(x_{k-1} + \alpha_{k-1} p_{k-1}) \leq f(x_{k-1}) + c_1 \alpha_{k-1} g_{k-1}^T p_{k-1}, \tag{9}$$

$$g^T(x_{k-1} + \alpha_{k-1} p_{k-1}) p_{k-1} \geq c_2 g_{k-1}^T p_{k-1}, \tag{10}$$

where $p_{k-1} = -H_{k-1} g_{k-1}$ is the QN search direction, and $c_1 \in (0, 1)$ and $c_2 \in (c_1, 1)$ are some constants. Hence to assure $s_{k-1}^T \widetilde{y}_{k-1} > 0$, it is enough in the expression of $\widetilde{y}_{k-1}$ defined by (2), ϕ_{k-1} is replaced by

$$\phi_{k-1} = (\eta - 1) s_{k-1}^T y_{k-1} \quad \text{if} \quad \phi_{k-1} < (\eta - 1) s_{k-1}^T y_{k-1}, \tag{11}$$

where $\eta \in (0, 1)$, as suggested by Zhang and Xu [7]. This replacement yields the following inequalities

$$\begin{aligned} \phi_{k-1} &\geq (\eta - 1) s_{k-1}^T y_{k-1}; \\ \beta_{k-1} &\geq \eta; \\ s_{k-1}^T \widetilde{y}_{k-1} &\geq \eta s_{k-1}^T y_{k-1}. \end{aligned}$$

These relations are needed in the next section.

3 Convergence property

In this section, the global and local convergence properties of the modified limited memory BFGS algorithm are studied. Technically, the modified LBFGS matrix H_k is obtained by updating a bounded matrix, m times, using the MBFGS formula. For the purposes of our analysis, we shall use the direct update $B_k = H_k^{-1}$, which is the modified Hessian approximation to the objective function, associated with the modified Eq. (8).

```
Algorithm 1 (Modified limited memory BFGS)
Given an initial point $x_0$, $m > 0$, $0 < c_1 < c_2 < 1$, a symmetric positive definite initial matrix $B_0 = B_0^{(0)}$;
$k \leftarrow 1$;
repeat
    Compute the search direction $p_{k-1} \leftarrow -B_{k-1}^{-1} g_{k-1}$;
    Choose a steplength $\alpha_{k-1}$ along the direction, to satisfy the Wolfe conditions (9) and (10), and set
        $x_k = x_{k-1} + \alpha_{k-1} p_{k-1}$;
    If $k > m$, discard the vector pair $\{s_{k-m-1}, y_{k-m-1}\}$ from storage;
    Compute and save $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$;
    Quantify the value of $\phi_{k-1}$ via (2) and (11);
    Define $\beta_{k-1} = 1 + \phi_{k-1}/s_{k-1}^T y_{k-1}$;
    Update $B_{k-1}^{(0)}$, $m$ times by using the pairs $\{s_i, y_i\}_{k-m}^{k-1}$ as:

    for $l = 0$ to $m - 1$

        $B_{k-1}^{(l+1)} = B_{k-1}^{(l)} - \frac{B_{k-1}^{(l)} s_{l+k-m} s_{l+k-m}^T B_{k-1}^{(l)}}{s_{l+k-m}^T B_{k-1}^{(l)} s_{l+k-m}} + \frac{\beta_{l+k-m} y_{l+k-m} y_{l+k-m}^T}{y_{l+k-m}^T s_{l+k-m}};$   (12)

    end (for)
    Set $B_k = B_{k-1}^{(m)}$;
    $k - 1 \leftarrow k$;
until termination
```

For the purposes of this section, we make the following assumptions on the objective function.

Assumption 1 *(a) $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is a convex set.*

(b) f is twice continuously differentiable.

(c) f is uniformly convex function, i.e. there exist positive constants M_1 and M_2 such that

$$M_1 \|z\|^2 \leq z^T G(x) z \leq M_2 \|z\|^2, \tag{13}$$

for all $z \in \mathbb{R}^n$ and all $x \in \Omega$.

In the sequal we show that the new algorithm is globally convergent, and its convergence rate is R-linear. The proof of Theorem 1 is an extension to that of [4].

Theorem 1 *Let x_0 be a starting point, for which f satisfies Assumptions 1, and suppose matrices $B_k^{(0)}$ are symmetric positive definite initial matrices, for which $\{\| B_k^{(0)} \|\}$ and $\{\| B_k^{(0)^{-1}} \|\}$ are bounded. Then, the sequence $\{x_k\}$ generated by Algorithm 1 converges to the unique x^* on Ω, and the convergence rate is R-linear, that is, there is a constant $0 \leq r < 1$, such that*

$$f(x_k) - f(x^*) \leq r^k (f(x_0) - f(x^*)). \tag{14}$$

Proof Using the definition of $\bar{G}_{k-1} = \int_0^1 G(x_{k-1} + \tau s_{k-1}) d\tau$, we have

$$y_{k-1} = \bar{G}_{k-1} s_{k-1}. \tag{15}$$

Then Eqs. (13) and (15) give

$$M_1 \|s_{k-1}\|^2 \leq y_{k-1}^T s_{k-1} \leq M_2 \|s_{k-1}\|^2, \tag{16}$$

and

$$\frac{\|y_{k-1}\|^2}{y_{k-1}^T s_{k-1}} = \frac{s_{k-1}^T \bar{G}_{k-1}^2 s_{k-1}}{s_{k-1}^T \bar{G}_{k-1} s_{k-1}} \leq M_2. \tag{17}$$

Moreover, because of the second order Taylor expansion, we have

$$\begin{aligned} |\phi_{k-1}| &= |4(f(x_{k-1}) - f(x_k)) + 2(g_{k-1} + g_k)^T s_{k-1}| \\ &\leq 2|s_{k-1}^T G(x_{k-1} + ts_{k-1})s_{k-1} + (g_k - g_{k-1})^T s_{k-1}|. \end{aligned} \tag{18}$$

Now, from (18), (16) and (13), we obtain

$$|\phi_{k-1}| \leq 4M_2\|s_{k-1}\|^2. \tag{19}$$

Since the strategy (11) implies $\phi_{k-1} \geq (\eta - 1)s_{k-1}^T y_{k-1}$, from (16) and inequality (19), we have

$$\eta \leq |\beta_{k-1}| = \left|1 + \frac{\phi_{k-1}}{s_{k-1}^T y_{k-1}}\right| \leq 1 + \frac{4M_2}{M_1}. \tag{20}$$

Subsequently, let $B_k^{(0)} = (H_k^{(0)})^{-1}$, and using the arguments in [5], the following inequality holds.

$$trace\left(B_{k-1}^{(j)}\right) \leq trace\left(B_{k-1}^{(j-1)}\right) + \left(1 + \frac{4M_2}{M_1}\right) M_2, \qquad j = 1, 2, \cdots, m.$$

Since $B_{k-1}^{(0)}$ is bounded, we obtain

$$trace(B_k) = trace\left(B_{k-1}^{(m)}\right) \leq trace\left(B_{k-1}^{(0)}\right) + m\left(1 + \frac{4M_2}{M_1}\right) M_2 \leq M_3, \tag{21}$$

for some positive constant M_3. Analogically, we can also have a simple expression for the determinant as well, see [4].

$$\begin{aligned} det(B_k) = det(B_{k-1}^{(m)}) &= det(B_{k-1}^{(0)}) \prod_{l=1}^{m} \frac{\widetilde{y}_l^T s_l}{s_l^T B_{k-1}^{(l)} s_l} \\ &= det(B_{k-1}^{(0)}) \prod_{l=1}^{m} \frac{\widetilde{y}_l^T s_l}{s_l^T s_l} \frac{s_l^T s_l}{s_l^T B_{k-1}^{(l)} s_l} \\ &= det(B_{k-1}^{(0)}) \prod_{l=1}^{m} \beta_l \frac{y_l^T s_l}{s_l^T s_l} \frac{s_l^T s_l}{s_l^T B_{k-1}^{(l)} s_l}. \end{aligned}$$

Since by (21), the largest eigenvalue of $B_{k-1}^{(l)}$ is also less than M_3, using the left hand side of (16), the boundedness of $\|(B_{k-1}^{(0)})^{-1}\|$, and inequality (20), we have

$$det(B_k) \geq det\left(B_{k-1}^{(0)}\right)\left|\prod_{l=1}^{m}\beta_l\right|\left(\frac{M_1}{M_3}\right)^m \geq M_4, \tag{22}$$

for some positive constant M_4.

Hence, by (21) and (22), there is a constant $\delta > 0$ such that

$$\cos\theta_{k-1} = \frac{s_{k-1}^T B_{k-1} s_{k-1}}{\|s_{k-1}\|\|B_{k-1}s_{k-1}\|} \geq \delta. \tag{23}$$

Moreover, the Wolfe conditions (9, 10) and Zoutendijk [6] theorem imply that

$$\sum_{k=1}^{\infty}\cos^2\theta_{k-1}\|g_{k-1}\|^2 < \infty. \tag{24}$$

Since $\cos\theta_{k-1}$ is bounded away from zero for all $k \geq 1$, then the sequence $\{x_k\}$ converges to x^* in the sense that

$$\lim_{k\to\infty}\|g_{k-1}\| = 0. \tag{25}$$

Now, because of the Wolfe conditions (9) and (10), and Assumptions 1, it can be shown (see [4] for details) that there is a constant $c > 0$, in which

$$f(x_k) - f(x^*) \leq (1 - c\cos^2\theta_{k-1})(f(x_{k-1}) - f(x^*)). \tag{26}$$

Applying (23) to the last inequality, gives rise to the conclusion (14).
Moreover, using (13) and Taylor expansion we have

$$\frac{1}{2}M_1\|x_k - x_*\|^2 \leq f(x_k) - f(x^*), \tag{27}$$

which together with (14) gives $\|x_k - x_*\| \leq r^{k/2}[2(f(x_0) - f(x^*)/M_1]^{1/2}$, implying that the convergence rate is R-linear.

4 The algorithm

In this section, we give a description of our ”Modified Limited memory BFGS ”, (MLBFGS) method. It is well known that the ordinary limited memory BFGS, (LBFGS) method, proposed by Liu and Nocedal [4], is adopted from the BFGS method to solve large scale problems. Very often, the QN methods can employ a scaled identity as an initial approximation to speed up the convergence. For this purpose, Liu and Nocedal [4] suggested the use of $H_{k-1}^{(0)} = \gamma_{k-1}I$, where $\gamma_{k-1} = y_{k-1}^T s_{k-1}/\|y_{k-1}\|^2$.

To explain the strategy of the limited memory BFGS method, we first assume that m is a given small positive integer. The LBFGS method keeps the m most recent correction pairs, $\{s_i, y_i\}_{i=k-m-1}^{k-2}$, to update m times, the matrix $H_{k-1}^{(0)}$, and computes the product $H_{k-1}g_{k-1}$ efficiently via performing two-loop recursion procedure, see [4,5]. To modify ordinary limited memory BFGS method, we require to redefine y_i via multiplying it by β_i for $i = k-m-1, \ldots, k-2$. Therefore, our two loop recursion procedure of the MLBFGS method, for computing the product $H_{k-1}g_{k-1}$, is given as follows:

Algorithm 2 *(MLBFGS two loop recursion)*

```
$q \leftarrow g_{k-1};$
For  $i = k-2, k-3, \cdots, k-m-1$
    Compute $\tilde{\rho}_i = \frac{1}{\beta_i y_i^T s_i};$
    $\alpha_i \leftarrow \tilde{\rho}_i s_i^T q;$
    $q \leftarrow q - \alpha_i y_i;$
end(for)
$r \leftarrow H_{k-1}^{(0)} q;$
For  $i = k-m-1, k-m-2, \cdots, k-2$
    $\chi \leftarrow \tilde{\rho}_i y_i^T r;$
    $r \leftarrow r + s_i(\alpha_i - \chi);$
end(for)
Stop with result $r = H_{k-1}g_{k-1}$.
```

5 Numerical experiments

This section evaluates the performance of the MLBFGS algorithm in comparison with the standard LBFGS algorithm of Liu and Nocedal [4]. Both codes are written in Fortran 77 in double precision. A line search routine is used based on the quadratic and cubic interpolations that satisfies the strong Wolfe conditions (9) and

$$|g^T(x_{k-1} + \alpha_{k-1}p_{k-1})p_{k-1}| \leq c_2|g_{k-1}^T p_{k-1}|, \tag{28}$$

with $c_1 = .01$ and $c_2 = .9$, see for example Fletcher [3].

we considered fifty large-scale unconstrained optimization test problems with the standard starting points selected from [1]. For each test problem, ten numerical experiments with dimension ranging from 1000 to 50,000 were performed. Extensive numerical testing of the methods are carried out by selecting three values for m, i.e. 3, 5 and 7. As compared with the results of these values of m in the two limited memory BFGS algorithms, it is observed that the choice of the parameter m=5 gives the best result. To maintain positive definiteness of the curvature approximation, strategy (11) is employed whitin MLBFGS algorithm with $\eta = 10^{-4}$. All runs were terminated when

$$\|g_k\| \leq 10^{-5}\max\{1, \|x_k\|\}.$$

In the numerical comparisons we selected to use the performance profiles of Dolan and Moré [2] to present the results of the numerical experiments. The performance pro-

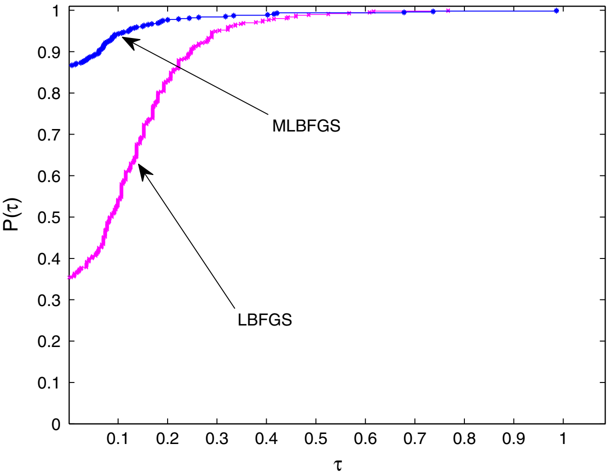

Fig. 1 performance profiles based on function and gradient evaluations

files correspond to the number of function–gradient evaluations metric. It is observed from Fig. 1 that the MLBFGS is always the top performer for all values of τ.

We observe that the amount of additional storage and arithmetics of MLBFGS algorithm are insignificant in comparison with those of the LBFGS method. The numerical result obtained reveals that a higher order accuracy in approximating the inverse Hessian matrix of the objective function takes place for the modified LBFGS method which enables it to outperforms the ordonary LBFGS method.

6 Conclusion

In this paper, we have explored a fourth order tensor model of the objective function to modify QN equation with the aim of attaining more accurate Hessian estimate. This enables us to design a more effective limited memory BFGS method to solve large-scale function minimization.

The major advantages of proposed algorithm are that it preserves the convergence properties of the LBFGS method, while allowing to obtain an overall computational saving, and it can be built very inexpensively.

Our experiments in the frame of LBFGS algorithm, showed that the MLBFGS is superior to the LBFGS algorithm.

Acknowledgments We would like to acknowledge respective referee for his useful suggestions and comments on the previous versions of this paper, which improve this paper significantly

References

1. Andrei, N.: An unconstrained optimization test functions collection. Adv. Model. Optim. **10**, 147–161 (2008)
2. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–203 (2002)
3. Fletcher, R.: Practical Methods of Optimization, 2nd edn. Wiley, New York (1987)

4. Liu, D., Nocedal, J.: On the limited memory BFGS method for large-scale optimization. Math. Program. **45**, 503–528 (1989)
5. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer Series in Operations Research, 2nd edn. Springer, New York (2006)
6. Sun, W., Yuan, Y.-X.: Optimization Theory and Methods Nonlinear Programming. Springer Optimization and Its Applications. Springer, New York (2006)
7. Zhang, J.Z., Xu, C.X.: Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations. J. Comput. Appl. Math. **137**, 269–278 (2001)
8. Zoutendijk, G.: Nonlinear programming, computational methods. In: Abadie, J. (ed.) lnteger and Nonlinear Programming, pp. 37–86. North-Holland, Amsterdam (1970)